Session 128

Font Management on Mac OS X

Nathan Taylor
Type, Software Engineer

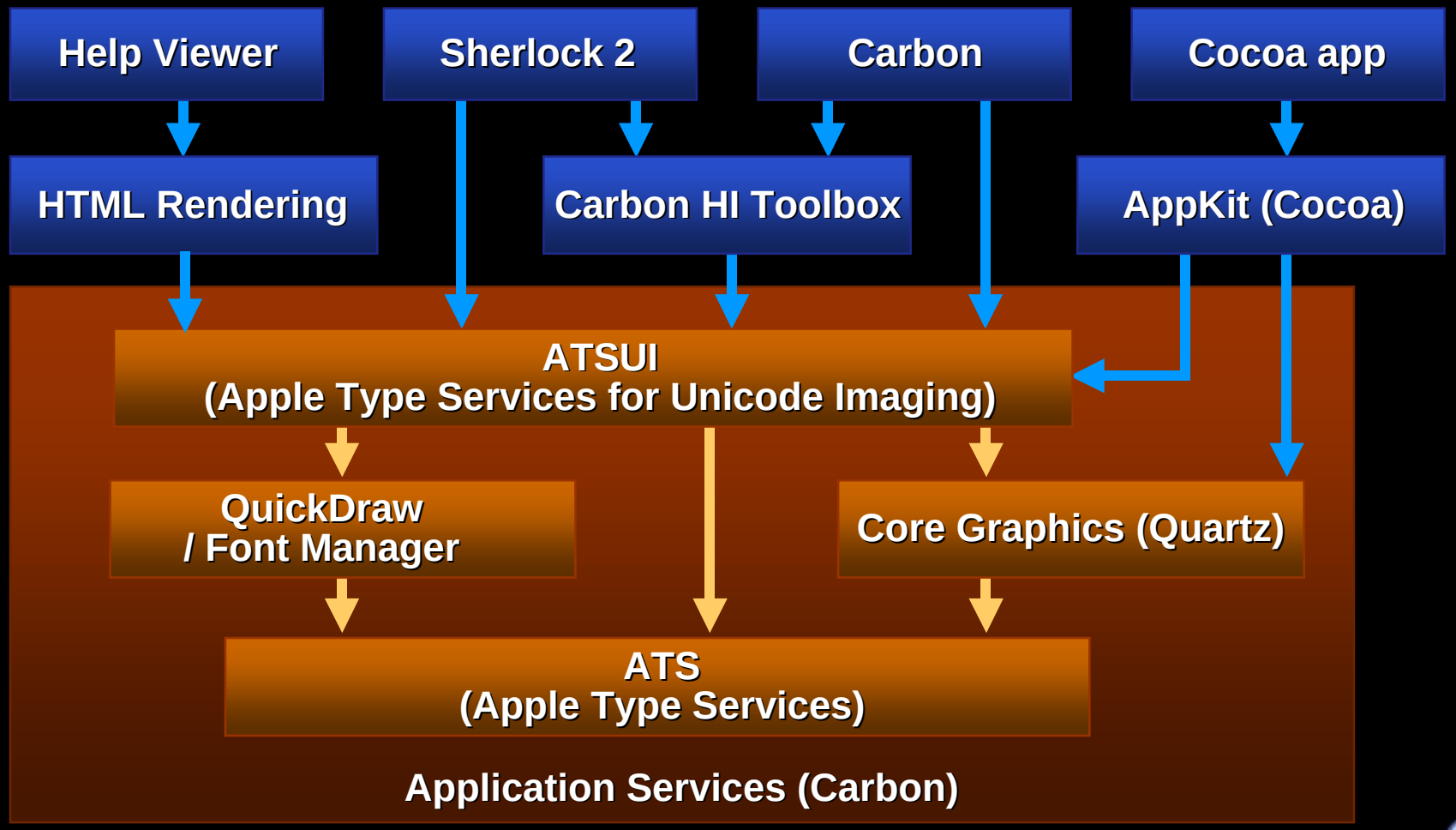
Agenda

- Inner workings
- Programming interface
- Real-life example

Enhancements

- Performance
- Scalability
- Extensibility (technologies/storage models)
- Consistent programming interface
- Non-invasive file access

ATS Client Hierarchy



Data Formats

- TrueType fonts
  - Font suitcases (resource, data/dfont)
  - Windows TrueType
- PostScript fonts
  - OpenType (CFF)
  - Type 1
  - Character Identifier-Keyed (CID)

Default Font Locations

- User: ~/Library/Fonts
- Local: /Library/Fonts
- Network: /Network/Library/Fonts
- System: /System/Library/Fonts
- Classic: System Folder/Fonts

Developer Font Locations

- Flexible: any directory
- Application library directory
- Application bundle
- Memory
- Application resource fork

Installing Fonts

- Drag and drop
- Directories scanned on application launch and user login
- Duplicates/conflicts resolution
  - Domain precedence
  - Synthesize family database entries
  - Font contents unmodified

Selecting Fonts

- Carbon font menu
  - Flat or hierarchical
- Cocoa font panel
  - Easy to select font, style, and size
  - Improved handling of large collections

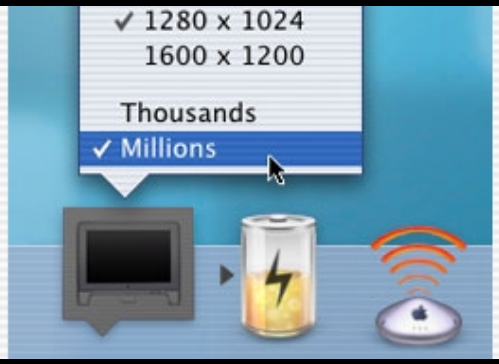
Printing

- Preview from any Carbon or Cocoa application
- PDF used as intermediary
- Fonts streamed via ATS
- Fonts embedded in document
- Cross-platform exchange

Demonstration Preview

- Standard font directories
- Selecting fonts in documents
- Printing and PDF
- Local font activation

Demo



Session 128

Programming Interface

Julio Gonzalez
Type, Senior Software Engineer

Programming Interface Topics

- API Sets
- Concepts
- Tasks
- Performance

Time Warp

- Font Manager Inside Mac Volume 1:
  - InitFonts
  - GetFontName
  - GetFNum
  - RealFont
  - SetFontLock
  - FMSwapFont(Advanced Routine)

API Sets

- Font Manager (FM..)
  - e.g., FMCreateFontFamilyIterator
  - Available on Mac OS 9, CarbonLib, and Mac OS X
- ATS (ATSFont..)
  - e.g., ATSFontIteratorCreate
  - Available on Mac OS X

Programming Concepts

- Font family references
- Font references
- Containers
- Generations

Font Access Data Types

- FMFont, ATSFontRef, ATSUFontID
- FMFontFamily, ATSFontFamilyRef

Programming Tasks

- Storing Font References
- Data access
- Enumeration
- Conversion and compatibility
- Activation/deactivation
- Selection

Task: Storing Font References

- Past:
  - Naughty apps store FMFontFamily
  - Store FontFamily name (FOND)
- Suggestions:
  - PostScript, combination Full & Unique
  - FontSync references

Task: Data Access

- Old style:
  - Resource Manager
  - Low memory globals

Task: Data Access

- Accessor functions:
  - FMGetFontTable
  - FMGetFontContainer
  - ATSFontGetPostScriptName
  - ATSFontGetHorizontalMetrics

Example: Data Access

- Getting the FOND:
  - FMGetFontFromFontFamilyInstance to get font
  - FMGetFontContainer to get fsSpec
  - Open resource fork (data resource fork)
  - FMGetFontFamilyName
  - Get1NamedResource

Task: Enumeration

- Iterators and applier functions
  - FMCreateFontIterator, FMCreateFontFamilyIterator
  - FMCreateFontFamilyInstanceIterator
  - ATSFontCreateIterator, ATSFontApplyFunction
- Iterator Filters
  - Font Technology, Container, Generation, Callbacks

Example: Enumeration

- Create an iterator to enumerate all the families in the local context:

```
status = FMCreateFontFamilyIterator(..., &iterator);
while (FMGetNextFontFamily (&iterator, &fam) == noErr ) {
  status = FMCreateFontFamilyInstanceIterator(fam, &instl);
  ...
  status = FMDisposeFontFamilyInstanceIterator(&instl);
}
status = FMDisposeFontFamilyIterator(&iterator);
```

Task: Conversion

- Conversion APIs
  - FMGetFontFamilyInstanceFromFont
  - FMGetFontFromFontFamilyInstance

Task: Compatibility

- Compatibility APIs
  - ATSFontFamilyFindFromQuickdrawName
  - ATSFontFamilyGetQuickdrawName
  - ATSFontGetFileSpecification

Task: Activation/Deactivation

- Use ATS database functions
  - FMActivateFonts, FMDeactivateFonts
  - ATSFontActivateFromFileSpecification
  - ATSFontDeactivate
  - ATSFontActivateFromMemory

Example: Activation/ Deactivation

- Activate the fonts in a file:

```
ATSFontContainer container;
ATSFontRef      fontArray[n];

status = ATSFontActivateFromFileSpecification(
   &appFileSpec, kATSFontContextLocal, ...,
   &container);
status = ATSFontFindFromContainer(
   container, ..., fontArray, &fontCount);
```

Task: Selection

- Use the standard font menu functions
  - CreateStandardFontMenu
  - UpdateStandardFontMenu
  - GetFontFamilyFromMenuSelection

Example: Selection

- Create a basic font menu:

```
menu = GetMenuHandle(kFontMenuResID);
status = CreateStandardFontMenu(
                menu,
                CountMenuItems(menu),
                kFontMenuFirstHierMenuID,
                kHierarchicalFontMenuOption,
                &hierarchicalMenuCount);
```

Example: Selection

- Find the font family and style from the menu item selected by the user:

```
UpdateStandardFontMenu(GetMenuHandle(menuID),
    &gFontMenuCount );
menuResult = MenuSelect(event->where);
...
if ( menuID == kFontMenuResID )
    status = GetFontFamilyFromMenuSelection(
                        GetMenuHandle(menuID),
                        menuItem,
                        &fontFamily,
                        &style);
```

Performance

- ATSServer messaging
- Font file iteration
- Use APIs whenever possible

Session 128

Programming Example

Scott Thompson
Macromedia, Senior Software Alchemist

Text Handling

- Mid-range text handling capabilities
  - Columns
  - Justification
  - Custom kerning and tracking adjustments
- Text in a path
- Text along a path
- Text flowing around a path
- PostScript output

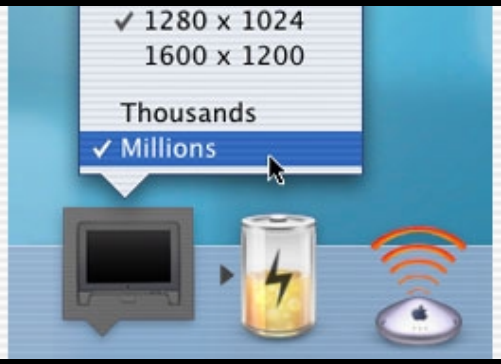
Font Family Resource (FOND)

- Font names (in particular PostScript name of font)
- Glyph widths and other glyph metrics
- Glyph reencoding tables and glyph names
- Kerning pair information

Glyph Outlines

- Our own PostScript Type 1 font parser
- Our own TrueType font parser
- Adobe Type Manager back end
- ATSUI Low Level programming interface

Wrap Up



Summary

- Remove Resource Manager dependencies
- Begin developing to the programming interface of Font Manager on Carbon
- Preview the programming interface of ATS Framework on Mac OS X

Roadmap

113 Text on Mac OS X	Hall 2 **Wed., 3:30pm**
122 Using Cocoa	Hall 2 **Thurs., 3:30pm**
FF105 Feedback Forum: International Technologies	Room C **Thurs., 5:00pm**

Documentation

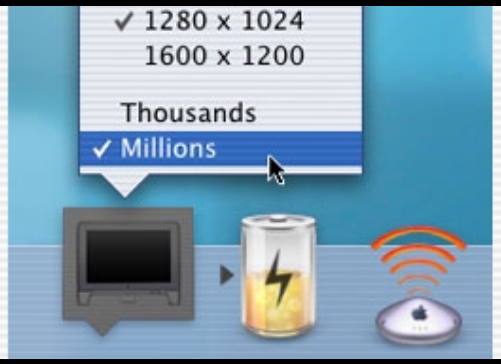
- Font Manager Technical Documentation
  http://developer.apple.com/techpubs/macosx/Carbon/text/FontManager/fontmanager.html
- FontSync Technical Documentation
  http://developer.apple.com/techpubs/macosx/Carbon/text/FontSync/fontsync.html

Who to Contact

Xavier Legros
Text & International Technology Manager
Apple Worldwide Developer Relations
xavier@apple.com

http://developer.apple.com/events/urls.html

Q&A



Xavier Legros
Text & International Technology Manager
xavier@apple.com

http://developer.apple.com/events/urls.html

WWDC
Worldwide Developers Conference 2001